Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

Display: 10 **Documents in Display Format:** - **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Thursday, December 11, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L14	L13 and (database or data with base or "data base")	22	L14
L13	L12 and (master or main or primary or controlling) same symbol	24	L13
L12	L11 and (stock near3 symbol or ticker near3 symbol)	92	L12
L11	securities near2 exchange	1164	L11
L10	705/37	1873	L10
L9	705/36	1260	L9
L8	705/35	1760	L8
L7	705.clas.	24228	L7
L6	704/10	615	L6
L5	704.clas.	16108	L5
L4	707.clas.	18023	L4
L3	707/100	3937	L3
L2	707/4	3238	L2
L1	707/1	5663	L1

END OF SEARCH HISTORY

☐ Generate Collection Print

L14: Entry 21 of 22 File: USPT Jan 21, 1986

US-PAT-NO: 4566066
DOCUMENT-IDENTIFIER: US 4566066 A

TITLE: Securities valuation system

DATE-ISSUED: January 21, 1986

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Towers; Frederic C.	Bethesda	MD	20034	

DISCLAIMER DATE: 19990608

APPL-NO: 06/ 385323 [PALM]
DATE FILED: June 4, 1982

PARENT-CASE:
CROSS-REFERENCE TO RELATED APPLICATION This invention is disclosed in a co-pending application, of which this application is a continuation, entitled, "Securities Valuation System, Ser. No. 279,781, filed Aug. 11, 1972, and issued as U.S. Pat. No. 4,334,270 on June 8, 1982.

INT-CL: [04] G06F 15/21

US-CL-ISSUED: 364/408
US-CL-CURRENT: 705/36

FIELD-OF-SEARCH: 364/408, 364/9MSFile

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	4064490	December 1977	Nagel	358/141 X
☐	4334270	June 1982	Towers	364/300

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
1447238	August 1976	GB	

OTHER PUBLICATIONS

Minker, J. et al., "File Organization and Data Management," Annual Review of Information Science & Technology, vol. 2, 1967, 123-160.

ART-UNIT: 236

PRIMARY-EXAMINER: Smith; Jerry

ASSISTANT-EXAMINER: Jablon; Clark A.

ATTY-AGENT-FIRM: Roberts & Floyd

ABSTRACT:

This discloses a system operating on a general purpose digital computer which produces securities portfolio valuation schedules for multiple simultaneous users. The computer maintains securities information in system-wide files which are updated, both electronically and manually, on a daily basis. The system permits each user to store information about his portfolios, as well as information about supplemental securities not contained in the system-wide files.

1 Claims, 1 Drawing figures

L14: Entry 21 of 22 File: USPT Jan 21, 1986

DOCUMENT-IDENTIFIER: US 4566066 A
TITLE: Securities valuation system

Brief Summary Text (23):
The Bunker-Ramo Corporation, among others, currently produces a daily computerized tape listing of the volume, high, low, and last price, as well as earnings, dividends, and other data for the securities on the following exchanges:

Brief Summary Text (25):
Of the 11,700 securities described above, there are about 30 changes of significance each day. These changes include stock splits, stock dividends, mergers, name changes, ticker symbol changes, new securities listings, transfers of securities from one exchange to another, or delistings.

Brief Summary Text (28):
In addition, each security is assigned a unique "ticker-symbol" comprising one to six characters, usually mnemonically related to the name of the security. This system of abbreviation is more commonly used than the other two by investors, registered representatives, traders, advisers, and "front-office" personnel because it is shorter and easier to remember.

Brief Summary Text (29):
Although both the Bunker-Ramo number and CUSIP number may be used internally, the "ticker symbol" is the customary external method of accessing securities pricing information in any quotation system, and in the printing quotation system making up the invention.

Brief Summary Text (34):
2. A comprehensive, reliable and accurate data base, carefully cross-indexed and cross-referenced.

Brief Summary Text (35):
3. User ability to add readily any security not contained in the data base.

Brief Summary Text (40):
The applicant is aware of no prior art system for producing individual portfolio valuation schedules from a computerized data base on a time-sharing system.

Brief Summary Text (41):
The application, broadly, would appear deceptively simple in view of the readily accessible computerized listings, which would form the data base against which the customer portfolio transactions are processed.

Brief Summary Text (52):
(3) A centralized securities data base is available to all users at minimal cost.

Brief Summary Text (64):
All of the securities on the daily Bunker-Ramo tape listing are included in the Group 1 master security files. The closing price, dividends, earnings, and other data for these securities are updated on a daily basis.

Brief Summary Text (67):
In producing a portfolio valuation schedule, the operator need only enter the ticker symbol for the Group 1 securities, the ticker symbol and the lastest price for the Group 2 securities and the Group 3 supplemental securities in order to produce a valuation schedule of the portfolio.

Brief Summary Text (70):
The accuracy of the reports is obviously greatly enhanced by the fact that all calculations are made automatically by a digital computer. Similarly, all formatting, sorting, alphabetizing and typing is machine-controlled. Accuracy is further insured by an extensive system of cross-reference files, which automatically adjust the valuation schedule to reflect mergers, stock dividends, name changes, ticker symbol changes and the like which may have occurred since the publication of the latest VALPORT Securities Index (for which I claim a copyright), or since the previous portfolio valuation.

Detailed Description Text (7):
The system operator communicates through his terminal 9 to the memory location of the priced securities 11 which are part of the memory of the main frame computer. The priced securities are updated daily by the tape 13. The system operator may edit the priced securities through the CUSIP routine 14 or through the ticker symbol routine 15.

Detailed Description Text (8):
The time sharing customer communicates with the system through his terminal 10. He may both access and edit one of his own accounts, user file 12. When a ticker symbol for a new security is entered, the old/new ticker symbol routine (16) checks if the symbol is valid, and updates it 17, to enter new securities with the current CUSIP or security number.

Detailed Description Text (14):
A new user, typically a brokerage firm, first creates 19 required files in his time-sharing library, then runs the **VALSETUP program. It would be desirable to have the program create the required files, but the SBC version of the BASIC language does not currently permit a program to create a file. This program allows the user to gain access to the VALPORT data base and to supply information about himself to the system.

Detailed Description Text (26):
The supplemental securities files contain the supplemental securities of a user that are not included in the VALPORT master securities data base available to all users. The programs described in this section are used to create and maintain these supplemental securities files.

Detailed Description Text (29):
In addition, the `MOVE` option creates a cross-reference trail of any security moved in the files. The Master System Operator can also use this program to shift a security within the Group 1 securities files, within the Group 2 securities files, or to move a security from Group 1 to 2, or from Group 2 to 1. Moreover, the Master System Operator can `merge` securities, making an entry in the ticker symbol cross-reference file **SYMX1 and the security number cross-reference file **OLDNEW#1. He can also change a ticker symbol and/or security number, making the appropriate cross-reference entries to the ticker symbol and security number cross-reference files, if necessary. An ordinary user can use the program in a more limited fashion to move securities within his supplemental securities file, or to cross-reference them to the Group 2 or Group 1 files.

Detailed Description Text (30):
The `CARD` option produces a 3.times.5 securities index card designed to help the

user keep track of the securities in his supplementary security files. The `FIND` option is a search routine used to retrieve and verify any security contained in either the user's supplemental security files or the VALPORT master securities data base.

Detailed Description Text (45):
Y01-Y12-Master Ticker Symbol to Security Number Cross-reference File

Detailed Description Text (46):
There are 12 of these files. Six are active at any given time and six are mirror-image copy files. The files are controlled and the active protected names appear in **CONTPORT and are accessed as Y$(1) through Y$(6). The sequence of Records is alphabetical by ticker symbol (characters 1-7) in ascending alphabetical sequence.

Detailed Description Text (47):
**SYMX1-Ticker Symbol Cross-reference Files

Detailed Description Text (48):
This is a symbol cross-reference file, **SYMX1, which is updated to SYMX2 and copied back. The Records are in ascending alphabetical order by old ticker symbol.

Detailed Description Text (59):
Alphabetically sorted ticker symbol files for every letter of alphabet. The purpose is to allow lookup of new security number rather than manually enter number. It is accessed in the **PRICESEC and **VALSEC (MOVE) programs.

Detailed Description Text (64):
X01-User Supplemental Security Ticker Symbol Files

Detailed Description Text (65):
The sequence of records is alphabetical by ticker symbol (characters 1-7) in ascending sequence.

Detailed Description Text (87):
The file CONTROL1 is then opened. In it is put an initial number (9,900,000) for the initial user supplemental security number, the names of each of the five portfolio files (P01 through P05), the names of the two user symbol ticker files (X01 and X02) and the names of the five user supplemental security files (S01 through S05). The file is then closed.

Detailed Description Text (89):
Each portfolio file (P01-P05), ticker symbol file (X01-X02), and supplemental security file (S01-S05) is opened and initialized by placing a numeric zero in the first field. Each file is then closed. Files CONTROL2 and TEMPO are opened as input files to test for their existence, and files OLDNEW#1 and OLDNEW#2 are opened as output files, initialized with numeric 0 in the first field, and closed. This procedure initializes all files and ensures that the user has made no error in establishing them.

Detailed Description Text (95):
The **CONTPORT file is opened and the four hexadecimal characters are read into the first four positions of the X$ array. The 30 security file control variables are then read into array M. The current date, monthly date, the highest supplemental security number, the alphabet, the numerals, the ticker symbol control file variable, the months of the year, the fractions, economic codes, and symbol and security file names are read and stored in the appropriate arrays and variables.

Detailed Description Text (102):
If the portfolio is to be entered from the keyboard, a sample line for formatting purposes is printed with or without a field for costs, depending on the answer to

the previous question. The user then types in the number of shares, the ticker symbol, and, optionally, customer's cost for each issue.

Detailed Description Text (105):
This routine alphabetically sorts the portfolio ticker symbols created above. A numeric array I(S) is built up in which the values in order correspond with the row number of the security in array U$(S)alphabetically sorted by symbol.

Detailed Description Text (107):
TICKER SYMBOL FILE SELECTION ROUTINE

Detailed Description Text (108):
The program then compares each ticker symbol in sequence to see if it is less than +G, +O, A, G, O, or 99 (A$(4) of **CONTPORT) to determine the proper ticker symbol to security number cross reference file (**Y$1-6).

Detailed Description Text (109):
For example, the ticker symbol comparison for IBM will set variable L equal to 5 since that ticker symbol will be in the fifth master ticker symbol to security number cross-reference file. Actually, that ticker symbol will be either in file Y05 or Y11 depending on which is the then currently active file. The variable Y$(5) from the previously opened **CONTPORT file will identify the correct file so that the statement "OPEN 1, Y$(L), INPUT" will open the correct file Y05 or Y11 to retrieve IBM.

Detailed Description Text (110):
TICKER SYMBOL DATA RETRIEVAL ROUTINE

Detailed Description Text (111):
The correct ticker symbol file is then opened. This file contains the SYMBOL/SECURITY NUMBER variables alpha-sorted by symbol, e.g.:

Detailed Description Text (112):
If the file contains over 20 records, the first 20 variables are read into F$(1) . . . F$(20). The 20th is compared with the desired ticker symbol. If the desired ticker symbol is greater, 20 more are read successively until the 20th is equal to or greater than the desired ticker symbol. (If the file contains 20 items or less, the program reads each one of the records individually.)

Detailed Description Text (113):
When 20 are to be searched, or when the last group of 20 or less is reached, each ticker symbol in the group is compared with the desired ticker symbol.

Detailed Description Text (114):
When found, the variable A$ contains the ticker symbol in the positions 1 through 7 and the security number in alpha form in positions 12 through 18. If no match is found, the **SYMX1 cross-reference file is searched for a new ticker symbol.

Detailed Description Text (115):
OLDNEW TICKER SYMBOL LOOKUP ROUTINE

Detailed Description Text (116):
If during the ticker symbol retrieval routine, the symbol being sought is not found, the program then branches to this routine. File **SYMX1 is opened. This file contains a string of records, each of which has the outdated ticker symbol and its corresponding newer or current ticker symbol. In either case, when a match is found, the program then branches back to the ticker symbol file selection routine and then the ticker symbol data retrieval routine. If, in this process, the new ticker symbol is not found, perhaps because it has again been modified, the program again goes back to this OLDNEW routine and looks for yet another current ticker

symbol corresponding to the newer ticker symbol.

Detailed Description Text (117):
If the ticker symbol sought is not found anywhere, the terminal prompts the user to enter a corrected ticker symbol or to enter a zero instructing the system to bypass the incorrect security.

Detailed Description Text (168):
If the old security number is found, then the new security number and exchange rate are determined from the cross-reference table. The routine calculates the new amount of the holding by multiplying the old holding times the exchange rate and adjusting any previous price. In the event the amount remains constant, the exchange rate is one.

Detailed Description Text (191):
Otherwise, the terminal prints each ticker symbol in sequence by security name and allows the operator to enter the total cost for each security. The program then changes the portfolio type, decreasing it by one, to indicate it contains cost information.

Detailed Description Text (201):
The operator is then asked if he wishes to record any sales. If so, he is prompted to enter the amount (shares or $ bonds), ticker symbol, and (optionally), the cost of the securities sold. The portfolio data is adjusted accordingly.

Detailed Description Text (202):
When all sales have been recorded, the operator is asked if he wishes to enter any purchases. If so, he is prompted to enter the amount, ticker symbol, and (optionally), the cost. For each such security, the system checks to see if the security is held in the portfolio. If so, the data is adjusted accordingly. If not, the system branches to the Ticker Symbol File Selection Routine and Ticker Symbol Data Retrieval Routine to find the security number; the Master Security File Selection Routine and Master File Data Retrieval Routine to retrieve the appropriate data for each such purchase.

Detailed Description Text (222):
For a death occurring during the week, the schedule must reflect the mean price of the high and low price (for exchange securities) or the bid and offer (for over-the-counter securities) and, for mutual funds, the offer price minus any volume discount.

Detailed Description Text (227):
A standard mutual fund discount schedule is next read, and control branches to the alpha Tag Sort Routine by ticker symbol, the Ticker Symbol File Selection Routine, the Ticker Symbol Data Retrieval Routine, the Alphabetic to Numeric Conversion Routine, followed by a numeric tag sort routine based on security number.

Detailed Description Text (238):
Each of the records in the Bunker-Ramo files A$(1)-A$(17), should match a corresponding record in the VALPORT files, M$(11)-M$(25), for the records in both sets of files are sorted in security sequence. If the ticker symbols match, the program puts into the daily price field of the VALPORT record the price variable appearing in one of the following fields (in the preference stated) from the Bunker-Ramo files: namely, the last price, close price (yesterday's last price), bid price, or asked price, as well as any new dividend, earnings, stock dividend date and stock dividend amount. (If the update is taking place on the day following the close of business of the first day of a new month, the old month-end price is discarded, and the previous day's closing price is transferred to the month-end price field before the new daily price is copied.)

Detailed Description Text (239):
On any given day, there will normally be about 9,000 matches and from 10 to 40 exceptions. These exceptions will be: a new security added to the Bunker-Ramo data, a security deleted from the Bunker-Ramo data, a security moved in the Bunker-Ramo file, or a security with a new ticker symbol or name. The VALPORT files completely parallel the Bunker-Ramo files with one exception at present. Stocks on the Pacific Coast Stock Exchange (which has about 50 stocks exclusively listed there and about 1,000 stocks also listed on the New York or American Exchange) are included in VALPORT only if they are exclusively listed on PCSE. Moreover, since the Bunker-Ramo data contains only zeros in the fields associated with bond interest rates an maturities, the correct data stored in VALPORT is not disturbed.

Detailed Description Text (266):
The program prompts the operator to enter a ticker symbol, checks that it contains 6 characters or less, and (for the master user) does not begin with "=".

Detailed Description Text (268):
When the ticker symbol supplied by the master operator is "+,,," or the security supplied by the user is "=...", representing Group 2 or Group 3 securities, the program automatically assigns a security number based on the latest Group 2 or Group 3 number.

Detailed Description Text (272):
Control then branches to the Ticker Symbol File Selection Routine, where the proper file, Y$(1-12) or Y$(7-8), is selected and opened for input together with its backup for output.

Detailed Description Text (273):
The first control field is increased by 1 to indicate the addition of a new security. Control then branches to the Ticker Symbol Data Retrieval Routine in which the contents of the old ticker symbol file are merged with the new ticker symbol and copied into the backup file.

Detailed Description Text (274):
Control then branches to the Master Security File Selection Routine and the Master Security Data Retrieval Routine in which the old file is merged together with the new security record and copied into the backup file. The File Name Flip-Flop Routine for both the security and ticker symbol files is then performed simultaneously so that the new record will be activated simultaneously in the ticker symbol file and master securities file. The master control file is then rewritten.

Detailed Description Text (275):
The meter is then adjusted to reflect the addition of a new security and ticker symbol. Control then returns to prompt the entry of another new ticker symbol. If the symbol is zero, the program ends.

Detailed Description Text (277):
This program is used to retrieve data about individual securities from the master security file when queried with the ticker symbol.

Detailed Description Text (279):
The user is prompted to enter a ticker symbol, and program control then branches to the Ticker File Selection Routine, the Ticker Symbol Data Retrieval Routine and, when the ticker symbol is found, the Alpha to Numeric Conversion Routine for the security number. The terminal then prints the ticker symbol and security number and goes through the Master Security File Selection Routine, the Master Security Data Retrieval Routine, and prints out the full name, economic code, dividend or interest, earnings or maturity, daily price and month-end price. The program then branches back and asks for another symbol. If it is zero, the program terminates.

Detailed Description Text (282):
The program begins with the Control File Variable Retrieval Routine. The operator is then prompted to enter one or a series of ticker symbols. The program then branches to the Pause Routine to permit the operator to insert a card.

Detailed Description Text (283):
After the card is inserted, the program indexes it, branches to the Ticker Symbol File Selection Routine, the Ticker Symbol Data Retrieval Routine, the Alpha to Numeric Conversion Routine, the Master File Selection Routine, and the Master Security Data Retrieval Routine. The card is typed with the name, date, ticker symbol, security number, industry code, price, and dividends and earnings or interest and maturity. If no further cards are to be printed, the program terminates.

Detailed Description Text (286):
The program begins with the Control File Variable Retrieval Routine. The operator is then prompted to supply a ticker symbol. It is checked to see if it begins with "=", indicating an error condition for the master operator.

Detailed Description Text (287):
Control branches to the Ticker File Selection Routine and the Ticker Symbol Data Retrieval Routine.

Detailed Description Text (288):
In a routine similar to that of **VALPORT(DEL) the ticker symbol file is copied over to its backup while blocking out the ticker symbol to be deleted.

Detailed Description Text (289):
In this process the initial numeric field is rewritten to reflect one less ticker symbol in the file.

Detailed Description Text (292):
After these procedures, a tandem File Name Flip-Flop Routine for the ticker symbol file names and the security file names is performed to allow simultaneous updating of the new files.

Detailed Description Text (293):
Following the deletion of the records and the file name Flip-Flop, the user's CONTROL1 or the master operator's CONTPORT file is rewritten. The METERA file is also updated to reflect the deletion of a security, and a message is printed, confirming the deletion of the security. The program then prompts the entry of another symbol. If it is zero, the program terminates.

Detailed Description Text (298):
Control then branches to the Ticker Symbol File Selection Routine, the Ticker Symbol Data Retrieval Routine, the Alpha to Numeric Conversion Routine, the Master Securities File Selection Routine and the Master Securities Data Retrieval Routine to locate the desired security.

Detailed Description Text (300):
Following the edit the new record is copied to the backup file, together with all of the remaining records in the security file. Control then branches to the File Name Flip-Flop Routine, and the control file is rewritten. The program then branches back to determine if there is a new ticker symbol to be edited. If a zero is entered, the program terminates.

Detailed Description Text (303):
This program is an abbreviated version of the VALSEC(EDIT) program using the same coding while by-passing certain sections. The user initially supplies a ticker

symbol and price for the security whose price he wishes to change.

Detailed Description Text (304):
The program then goes through the Control File Variable Routine, determines whether the ticker symbol begins with "=", goes to the Ticker Symbol File Retrieval Routine, the Ticker Symbol Data Retrieval Routine, the Alpha to Numeric Conversion Routine, the Master Security File Selection Routine, and the Master Security Data Retrieval Routine, copying over the records to the backup file with the edited data for the selected security. Next, the file name Flip-Flop Routine is executed, and the control file rewritten. If a ticker symbol of zero is entered, the program terminates.

Detailed Description Text (306):
This is a multi-purpose program that allows the VALPORT Master System Operator or any VALPORT user to move securities within the system to reflect changes of ticker symbols, mergers with other securities, movements from one exchange to another, or movements from Group 1, 2, or 3 to another Group.

Detailed Description Text (307):
A security may effectively be "moved" by a combined use of the **VALSEC `DEL` and `NEW` programs. The change should preferably be done through the **VALSEC `MOVE` program to insure a proper trail and cross-reference. In this way, a portfolio referring to the outdated ticker symbol or security number can be valued based on the current security data.

Detailed Description Text (309):
User Option 1. This program will change an existing user Group 3 security ticker symbol by deleting the old ticker symbol from the ticker symbol file, adding the new ticker symbol to the file, and noting the new ticker symbol in the proper field of that security record in the securities file. (The security number is not changed.)

Detailed Description Text (310):
User Option 2. The program will merge a Group 3 security with an existing Group 3 security by deleting the old ticker symbol, deleting the old security record, and adding a record to the OLDNEW#1 file showing the two security numbers and the exchange rate.

Detailed Description Text (311):
User Option 3. The program will merge a Group 3 security with either a Group 2 or Group 1 security by deleting the old ticker symbol, deleting the old security record, and adding a record to the OLDNEW#1 file, cross-referencing the system to the security in the VALPORT master files.

Detailed Description Text (312):
Master Option 1. The program will change an existing Group 2 security to a new Group 2 security by deleting the old ticker symbol, re-entering the new ticker symbol, noting the new ticker symbol field in the master security record, and putting a new record in the SYMX1 ticker-symbol-to-ticker-symbol cross-reference.

Detailed Description Text (313):
Master Option 2. The program will merge an existing Group 2 security into another existing Group 2 security by deleting the old ticker symbol, deleting the old security record, and putting a cross-reference trail in the SYMX1 and OLDNEW#1 file for ticker symbol and security number cross-reference.

Detailed Description Text (314):
Master Option 3. The program will transfer an existing Group 2 security to a new Group 1 security by deleting the old ticker symbol and security record, going to the **SYM#A-Z files for the new Bunker-Ramo security number, entering the new

ticker symbol and security record, and cross-referencing both in the SYMX1 and OLDNEW#1 file.

Detailed Description Text (315):
Master Option 4. The program will merge an existing Group 2 security into an existing Group 1 security by deleting the old ticker symbol, deleting the old security record, and cross-referencing both in the SYMX1 and OLDNEW#1 files.

Detailed Description Text (316):
Master Option 5. The program will change an existing Group 1 security to a new Group 1 security by deleting the old ticker symbol and security records, going to the **SYM#A-Z files for the new Bunker-Ramo security number, re-entering the new ticker symbol and security record and cross-referencing both with SYMX1 and OLDNEW#1 as necessary.

Detailed Description Text (317):
Master Option 6. The program will merge an existing Group 1 security into an existing Group 1 security by deleting the old ticker symbol and security records, and cross-referencing both in SYMX1 and OLDNEW#1.

Detailed Description Text (318):
Master Option 7. The program will merge an existing Group 1 security with an existing Group 2 security by deleting the old ticker symbol and security record and cross-referencing both in SYMX1 and OLDNEW#1.

Detailed Description Text (319):
Master Option 8. The program will change an existing Group 1 security to a new Group 2 securith by deleting the old ticker symbol and security record, re-entering the new ticker symbol and security record, and cross-referencing both in SYMX1 and OLDNEW#1.

Detailed Description Text (323):
The number of security records can be checked against the number of ticker symbol records to ascertain whether they are equal. This feature is used to verify a breakdown during editing to determine it is necessary to examine the contents of the files.

Detailed Description Text (324):
The CATSUPP option allows the user to print the contents of a cumulative supplement to the Securities Index supplement. The user may begin at any page or can enter page 0 to selectively print additions, deletions or changes, beginning with a certain letter or the name of a given security. This allows a user to ascertain if there has been any change in the ticker symbol or otherwise with respect to a particular security.

Detailed Description Text (327):
This program allows a backup of the VALPORT master securities files for data protection. The program begins with the Control File Variable Retrieval Routine, opens SECFILE1 for output and copies the first 15 securities files, M$(1)-M$(15), and closes SECFILE1. It then opens SECFILE2 for output and puts the security records from files M$(16) through M$(25) in that file and closes it. Lastly, it opens SYMFILE and puts into it each of the VALPORT master ticker symbol files. In each case, the control heading for the file is not copied.

Detailed Description Text (329):
This program recovers the backed-up security files and ticker symbol files by opening SECFILE1, SECFILE2, and determining (from the CONTPORT file control array), the highest security to be put in each new file. It counts the records in SECFILE1 and SECFILE2 to determine the number of securities to be put in each VALPORT master securities file.

Detailed Description Text (331):
Following that procedure, the program opens the SYMFILE and determines the number of ticker symbols to go into each of the VALPORT master ticker symbol files. It then goes through the appropriate data retrieval routine and copies the records in to the proper files with the appropriate control heading number.

Detailed Description Text (337):
The program begins with the Control File Variable Retrieval Routine and then opens CATFILE and CATREF as output files. It places the current pricing date in CATREF. Each of the VALPORT master security files is then opened in sequence. The security name, ticker symbol, security number, and classification code are read into CATFILE while the security numbers alone are read into CATREF. After all of the securities have been read, all files are closed, and program control branches to the **SORTOBJ program.

Detailed Description Paragraph Table (16):
FILE DESCRIPTIONS Name Description Page SYSTEM-WIDE FILES **VALPUSER User validation codes P-1 **CONTPORT VALPORT master control P-2 M$(1-50) Master security P-3 *Y01-Y06 Ticker symbol to security number P-4 cross-reference **SYMX1-2 OLD to NEW ticker symbol cross- P-5 reference **OLDNEW#1-#2 OLD to NEW security number P-6,7 cross-reference **GFINFILE Master control of Bunker-Ramo P-8 source files **ANYSE01-04 Bunker-Ramo source files (accessed P-9 **BASE001-3 once daily to update VALPORT **COTC001-3 files) **DNYBE01-04 **EPCSE01-02 **FMWSE01 CATFILE P-10 SUPFILE EXCPLIST Daily exception list P-11 **SYM#A-Z P-12 USER FILES CATALOG1 User control variables; represen- P-13 tatives initials CONTROL1 Control file for user files P-14 METERA Monthly usage report P-15 TEMPO Temporary portfolio storage P-16 X01, X02 User supplemental security ticker P-17 symbol/security number cross-reference S01, 02, 03 User supplementary securities P-18 04, 05 P01, 02, 03 User permanent portfolio storage P-19 04, 05 OLDNEW#1-#2 OLD to NEW security number cross- P-20 reference

Detailed Description Paragraph Table (18):
Field No. Field Code Field Description 1 X$(1) Hexadecimal buffer dump 2 X$(2) Hexadecimal end of transmission 3 X$(3) Hexadecimal upper case shift 4 X$(4) Hexadecimal lower case shift 5-34 M(30,1) Securities control number (lowest security in each of securities masterfiles) 35 D0 Daily price date (YYMMDD) 36 B0 Monthly price date (YYMMDD) 37 N0 Last Group 2 (+suppl) security number assigned 38 A$(1) ABCDEFGHIJKLM 39 A$(2) NOPQRSTUVWXYZ 40 A$(3) 0123456789 41 A$(4) +G+O A G 099 (control variable switch for finding proper ticker symbol file) 42-53 D$(1-12) Names of all months 54-95 E$(1-32) Names of all 1/32 fractions 96-145 I$,J$(1-25) Names of securities classification codes 146-151 V$(1-6) Ticker symbol file names 152-176 M$(1-25) Securities file names 177-188 Y$(7-12) Ticker rewind file name 189-213 M$(26-50) Securities rewind file name

Detailed Description Paragraph Table (19):
Field No. Field Code Field Description 1 N1 Number of records in file RECORD FIELDS (N1) 2,13 . . . S$ Security name (first 18 characters) 3,14 . . . T$ Security name (last 18 characters) 4,15 . . . U$ Ticker symbol 5,16 . . . N3 Security number 6,17 . . . C3 Classification code 7,18 . . . D3 Dividend/interest 8,19 . . . E3 Earnings/maturity 9,20 . . . V3 Daily price 10,21 . . . W3 Month-end price 11,22 . . . Y3 Stock dividend date 12,23 . . . Z3 Stock dividend amount

Detailed Description Paragraph Table (20):
File Contains Symbols

**Y01 +AAA - +FZZ (Group 2) **Y02 +GAA - +NZZ (Group 2) **Y03 +OAA - +ZZZ (Group 2) **Y04 AAA FZZ **Y05 GAA NZZ **Y06 OAA ZZZ Field No. Field Code Field Description 1 N1 Number of items in file RECORD FIELDS (N1) 2,3 . . . X$ Ticker symbol/security number Example of a Typical Record ##STR1## ##STR2## ##STR3## Symbol Blank Security number

Detailed Description Paragraph Table (21):
Field No. Field Code Field Description
1,2 . . . Q$ Old and new ticker symbols (End-of-file is marked by 0$="END**") Example of a Typical Record ##STR4## ##STR5## ##STR6## Old ticker Blank New ticker

Detailed Description Paragraph Table (24):
**ANYSE01-04 New York Stock Exchange **BASE001-003 American Stock Exchange **COTC001-003 Over-the-Counter (NASDAQ) **DNYBE01-04 New York Bond Exchange **EPCSE01-02 Pacific Coast Stock Exchange **FMWSE01 Midwest Stock Exchange & American Bond Exchange Field No. Field Code Field Description 1 A Security number 2 A$ Security name (begin) 3 B$ Security name (end) 4 C$ Ticker symbol (1-7) CUSIP (10-18) 5 B Clearinghouse number 6 C Open price 7 D High price 8 E Low price 9 F Last price 10 G Close price 11 H Adjusted close price 12 I Yearly high price 13 J Yearly low price 14 K Bid price 15 L Offer price 16 M Previous bid price 17 N Earnings indicator 18 O Quarterly earnings 19 P Annual earnings 20 Q Ex-dividend indicator 21 R Dividend payment indicator 22 S Stock dividend of indicator 23 T Quarterly dividend 24 U Annual dividend 25 V Stock dividend percent 26 W Cash dividend date 27 X Stock dividend date 28 Y Volume 29 Z Round lot indicator (End-of-file marked by A=0)

Detailed Description Paragraph Table (25):
CATFILE1 A reference file in security number sequence of next Securities Index. CATFILE Same as CATFILE1 but in alphabetical sequence. SECBOOK1 Printfile which controls high speed printer. CATFILE2 A reference file in Security Number Sequence of current published Securities Index. SUPPFIL1 Cumulative supplement file to CATFILE2 in numeric sequence. SUPPFILE SUPPFIL1 in alphabetical sequence. CATREF Contains date of CATFILE1 and Security Numbers contained in catfile. Field No. Field Code Field Description CATFILE1 (and CATFILE) 1,6 . . . S$ Name (begin) 2,7 . . . T$ Name (end) 3,8 . . . U$ Ticker symbol 4,9 . . . N3 Security number 5,10 . . . C3 Economic code CATFILE2 1 D5 DATE YYMMDD 2-6, 7-11 . . . S$,T$,U$,N3,C3 See CATFILE1 SUPPFIL1 (and SUPPFILE) 1-5, 13-17 . . . S$,T$,U$,N3,C3 See CATFILE1 6,18 . . . T3 Record type* 7-11, 19-24 . . . V$,W$,X$,W4,C4 Same fields for new record 12,24 . . . T4 Record type* EXCPLIST 1,6 . . . S$ Name (begin) 2,7 . . . T$ Name (end) 3,8 . . . U$ Ticker Symbol 4,9 . . . N Security Number 5,10 . . . C$ Blank for addition *** for deletion Ticker symbol if Ticker symbol change END** End of File **SYM#A-**SYM#Z 1,3 . . . A$ Ticker symbol 2,4 . . . N Security number
*Types: 1. Addition 2. Deletion 3. Old Record 4. New Record

Detailed Description Paragraph Table (28):
Field Field No. Code Field Description
1 X$ Portfolio shortname 2,3 N$,O$ Full account name 4 R8 Representative number 5 D8 Latest valuation date (YYMMDD) 6 D9 Previous valuation date (YYMMDD) 7 N2 Number of issues in portfolio 8 T2 Portfolio type 9 C2 Cash balance RECORD FIELDS (N2) 10,16 . . . P(,1) Security number 11,17 . . . P(,2) Classification code 12,18 . . . P(,3) Amount (shares or $ bonds) 13,19 . . . P(,4) Current price 14,20 . . . P(,5) Total customer cost 15,21 P(,6) Previous price (N2*6)+10,+19 S$( ) Security name (1st 18 characters) +11,+20 T$( ) Security name (last 18 characters) +12,+21 U$( ) Ticker symbol

+13,+12 D ( ) Dividend/interest +14,+23 E ( ) Earnings/maturity +15,+24 V ( ) Daily price +16,+25 W ( ) Month-end price +17,+26 Y ( ) Stock dividend date +18,+27 Z ( ) Stock dividend amount

Detailed Description Paragraph Table (29):

File Contains Symbols X01-02 =AAA - =ZZZ (Group 3) Field No. Field Code Field Description 1 N1 Number of items in file RECORD FIELDS (N1) 2,3 . . . X$ Ticker symbol/security number Example of a Typical Record ##STR7## ##STR8## ##STR9## Ticker symbol Blank Security number

Detailed Description Paragraph Table (30):

Field No. Field Code Field Description 1 N1 No. of records in file RECORD FIELDS (N1) 2,13 . . . S$ Security name (first 18 characters) 3,14 . . . T$ Security name (last 18 characters) 4,15 . . . U$ Ticker symbol 5,16 . . . N3 Security number 6,17 . . . C3 Classification code 7,18 . . . D3 Dividend/interest 8,19 . . . E3 Earnings/maturity 9,20 . . . V3 Daily price 10,21 . . . W3 Month-end price 11,22 . . . Y3 Stock dividend date 12,23 . . . Z3 Stock dividend amount

Detailed Description Paragraph Table (34):

File Selection Ticker Symbol File Selection Routine R-7 Master Security File Selection Routine R-10 Portfolio File Selection Routine R-12 Data Retrieval Ticker Symbol Data Retrieval Routine R-7 Master Security Data Retrieval Routine R-10 Control File Variable Retrieval Routine R-17 Portfolio Data Retrieval Routine R-12 Sorts Alpha tag sort routine R-6 Numeric tag sort routine R-9 Lookup OLD/NEW Ticker Symbol Lookup Routine R-8 OLD/NEW Security Number Lookup Routine R-21 Alpha to Numeric Conversion Routine R-9 Numeric to Alpha Conversion Routine Pause Routine R-15 Untangle Basic Date Routine R-5 Yes-No Routine R-5 File Name Flip-Flop Routine R-12

Detailed Description Paragraph Table (35):

Name Type Size Content M Numeric 30.times. 1 Securities file control variables F$ Alpha 20 Ticker symbol lookup array J Numeric 100.times. 6 Representative usage table X$ Alpha 8 Control variables A$ Alpha 5 Control constants X Numeric 165 Tag sort secondary array I Numeric 165 Tag sort primary array S$ Alpha 165 Security name 1st 18 characters T$ Alpha 165 Security name last 18 characters U$ Alpha 165 Ticker symbol N Numeric 165 Security number C Numeric 165 Classification code D Numeric 165 Dividend rate E Numeric 165 Earnings Z Numeric 165 Stock Dividend variables Y Numeric 165 Stock dividend date variables V Numeric 165 Month-end price P Numeric 165.times. 6 Portfolio data array

Detailed Description Paragraph Table (37):

INDEX TO PROGRAM LISTINGS Source Object Program Program User Name Name Name Program Function Page No.

DEMOPORT (DATAFILE) **DEMOPORT Typical portfolio with cost information BIGPORT (DATAFILE) **BIGPORT Large sample portfolio without cost information N72NEWP NEWPOBJ **VALPORT (NEW) Used to create a new portfolio; optionally store it T-1 to T-15 in master portfolio files and link to report options N72EDITP EDITPOBJ **VALPORT (EDIT) Used to edit a portfolio (including additions, deletions T-16 to T-33 and updating); optionally store it in master portfolio files and link to report options N72LISTP LISTOBJ **VALPORT (EDIT) Used by linking from EDITOBJ to list the edited portfolio N72DELP DELPOBJ **VALPORT (DEL) Used to delete a

portfolio from the master portfolio files N72MOVEP MOVEPOBJ **VALPORT (MOVE) Used to move the portfolio and change the portfolio shortname N72READ READOBJ **VALPORT (LIST) Used to produce a portfolio listing N72VAL VALPOBJ **VALPORT (VAL) Used to type the detailed valuation schedule, T-34 to T-47 **VALPORT (PUR) purchase recommendation schedule, or **VALPORT (SEL) sell recommendation schedule N72VALD VALDOBJ **VALPORT (VALD) Used to type the detailed valuation schedule without price/earnings ratio N72VALX VALPLOBJ **VALPORT (VALX) Used to type the extended valuation schedule showing gain and loss information N72SUM SUMPOBJ **VALPORT (SUM) Used to type the summary valuation schedule N72SUMPL SUMPLOBJ **VALPORT (SUMX) Used to type the extended summary valuation schedule N72TAX TAXPOBJ **VALPORT (TAX) Used to type the schedule of unrealized gains and losses (sorted by gain amount or gain percent) N72GAIN GAINOBJ **VALPORT (GAIN) Used to type the alphabetical schedule of gains and losses N72SIZE SIZEOBJ **VALPORT (SIZE) Used to type valuation schedule sorted by total value of holding ESTATE1 (DATAFILE) **ESTATE1 Sample portfolio for standard estate valuation ESTATE2 (DATAFILE) **ESTATE2 Sample portfolio for weekend estate valuation N72ESTP ESTPOBJ **VALPORT (EST) Used to type the standard estate valuation schedule N72ESTPW ESTPOBJW **VALPORT (EST) Used by linking from ESTPOBJ to type weekend estate valuation schedule N72XSP XSPOBJ **VALPORT (XS) First segment of user-customized access routine; retrieves portfolio and looks up securities information XSP (NONE) **XSP Second segment of user-customized access routine; links from XSPOBJ and used to weave and run (WVR) with user control programs to print user-customized reports XSDEMO (NONE) **XSDEMO Used to explain XS option to user N72COVP COVPOBJ **VALPORT (COV) Used to type the portfolio cover sheet N72HOLDS HOLDSOBJ **VALPORT (HOLDS) Used to list all portfolios holding a particular security N72SETUP SETUPOBJ **VALSETUP Used to gain access to the VALPORT system and T-48 to T-51 initially install the system in a particular user "library" VALPORT (NONE) **VALPORT Master control program for VALPORT T-52 to T-53 CHEKPORT (NONE) **CHEKPORT Determine location of errors in source data files USEPORT (NONE) **USEPORT Print detailed report of VALPORT system usage MGTPORT (NONE) **MGTPORT Print summary report of VALPORT usage, reset usage meter EDITCAT (NONE) **EDITCAT Add or edit representatives' initials, T-54 to T-58 representatives' initials and code numbers, edit forms control LISTCONT (NONE) LISTCONT Permits VALPORT master System Operator to access and print contents of VALPORT master control file EDITCONT (NONE) EDITCONT Permits VALPORT Master System Operator to edit contents of VALPORT master control file N72PRICE PRICEOBJ PRICESEC Used to automatically update VALPORT T-59 to T-63 securities files from raw Bunker-Ramo data and create exception file N72SUPGN SUPGNOBJ NONE Used by linking from PIRCEOBJ to compare current master securities files with reference file created at time of publication of latest securities index; exceptions automatically create a cumulative supplement which is stored in SUPPFIL1 SUPPSORT* SORTSUPP NONE Used to sort contents of SUPPFILE1 into alphabetical sequence storing results in SUPPFILE N72NEWS NEWSOBJ **VALSEC (NEW) Used by any user to enter security into user sup- T-64 to T-74 plemental files; used by VALPORT Master User to enter new security into VALPORT master securities files N72FINDS FINDSOBJ **VALSEC (FIND) Used to retrieve data about an individual security from a file when queried with ticker symbol N72CARDS CARDSOBJ **VALSEC (CARD) Used to produce a 3.times.5 card as a permanent record of securities stored in supplemental file N72DELS DELSOBJ **VALSEC (DEL) Used by any user to delete a security from his supplemental securities file or by VALPORT Master User to delete a security from VALPORT master securities files N72EDITS EDITSOBJ **VALSEC (EDIT) Used by any user to edit securities in user sup- plemental securities files and by VALPORT master user to edit securities in VALPORT master secu- rities files N72EPRI EPRIOBJ **VALSEC (EDITPRI) Used to update prices in user supplemental secu- rities files or VALPORT master securities file N72MOVE MOVEOBJ **VALSEC (MOVE) Move security records with automatic cross-indexing T-75 to T-86 N72CATS CATSOBJ **VALSEC (CAT, Provides count of securities masterfiles or CATSUPP,=SUPP) user supplemental security files; and prints cumulative supplement to latest VALPORT Securities Index, and prints alphabetical list of users supplemental securities BACKSEC NONE BACKSEC Backup securities masterfiles and ticker symbol masterfiles RECOSEC NONE RECOSEC Recover from BACKSEC N72SECD SECOBDS **SECDATES Print pricing dates for Bunker-Ramo and VALPORT files CATGEN NONE CATGEN

Strips security name, ticker symbol, security number and classification code from securities masterfiles and stores in CATFILE1 SORTSEC SORTOBJ NONE Alpha sort from CATFILE1 to CATFILE PRINTFCT NONE PRINTFCT Print contents of CATFILE or produce internal print file to produce VALPORT securities index CHEKSECS NONE CHEKSECS Print daily securities exception list from **EXCPLIST VALSEC NONE **VALSEC Master control program for VALSEC T-87rams SUPPSEC NONE **SUPPSEC Print alphabetized listing of Group 3 supplemental securities

Woven with SBC **CBSORT program to provide customized sort routine. ##SPC1## 15/962 20/962 25/926 30/962 35/962 40/666

First Hit Fwd Refs

Generate Collection Print

L14: Entry 18 of 22 File: USPT Dec 14, 1993

US-PAT-NO: 5270922
DOCUMENT-IDENTIFIER: US 5270922 A

TITLE: System for distributing, processing and displaying financial information

DATE-ISSUED: December 14, 1993

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Higgins; Gerard M.	Staten Island	NY		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Merrill Lynch & Company, Inc.	New York	NY			02

APPL-NO: 07/ 725951 [PALM]
DATE FILED: June 27, 1991

PARENT-CASE:
This is a continuation of copending application Ser. No. 06/626,339 filed on Jun. 29, 1984 now abandoned.

INT-CL: [05] G06F 15/20, G06G 7/52

US-CL-ISSUED: 364/408
US-CL-CURRENT: 705/37; 340/825.26

FIELD-OF-SEARCH: 364/408, 364/229.41, 364/283.2, 340/825.27, 395/800

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	3611294	October 1971	O'Neill et al.	340/825.26
☐	3792462	February 1974	Casey et al.	340/324
☐	3823387	July 1974	McClellan	340/825.27
☐	3911403	October 1975	O'Neill, Jr.	364/900
☐	3976840	August 1976	Cleveland et al.	340/825.26 X
☐	4008460	February 1977	Bryant et al.	364/200

☐	4063031	December 1977	Grunza	381/43
☐	4186438	January 1980	Benson et al.	364/200
☐	4253157	February 1981	Kirschner et al.	364/900
☐	4334270	June 1982	Towers	364/408
☐	4346442	August 1982	Musmanno	364/900
☐	4376978	March 1983	Musmanno	364/900
☐	4398250	August 1983	Hosono	364/900 X
☐	4432057	February 1984	Daniell et al.	364/300
☐	4473824	September 1984	Claytor	340/825.27 X
☐	4486853	December 1984	Parsons	364/408 X
☐	4491725	January 1985	Pritchard	235/375
☐	4547851	October 1985	Kurland	364/410
☐	4567359	January 1986	Lockwood	235/381
☐	4577062	March 1986	Hilleary et al.	179/2A
☐	4881179	November 1989	Vincent	364/518

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
1489571	October 1977	GB	364/408

OTHER PUBLICATIONS

"Technical Analysis Software Directory", Wall Street Computer Review, vol. 2, No. 8, Jun. 1985, 67-70, 72-79.
Everest, G. C. Database Management. (McGraw-Hill Book Company: New York) 1986, 746.

Chamoff, M. E. et al. "Dynamic Self-Optimizing Price Lookup for Retail Terminal System", IBM Tech. Discl. Bull, vol. 24, No. 2, Jul. 1981, 976-8.

ART-UNIT: 231

PRIMARY-EXAMINER: Envall, Jr.; Ray N.

ASSISTANT-EXAMINER: Brutman; Laura

ATTY-AGENT-FIRM: Hopgood, Calimafde, Kalil, Blaustein & Judlowe

ABSTRACT:

A data processing and communication system distributes and displays financial market ticker, quotation, news and ancillary information via a plurality of stored program controlled work stations. Stock trade executions, quotations and other ticker plant information is communicated in parallel to a hierarchy of system data processing terminals, e.g., those located at area, branch and individual work station locations. Storage media at the several system data processing levels extracts and stores data base information of differing purport and completeness for the disseminated data to support the system work station users.

In accordance with one aspect of the present invention, information characterizing a dynamically changing sub-population of the overall ensemble of market securities is maintained at and becomes immediately available to each work station responsive to the pattern of usage at that specific station. Various derivative tasks, such as security price limit alerts, are user programmable and are activated by the contents of the work station data base.

17 Claims, 5 Drawing figures

L14: Entry 18 of 22 File: USPT Dec 14, 1993

DOCUMENT-IDENTIFIER: US 5270922 A
TITLE: System for distributing, processing and displaying financial information

Abstract Text (1):
A data processing and communication system distributes and displays financial market ticker, quotation, news and ancillary information via a plurality of stored program controlled work stations. Stock trade executions, quotations and other ticker plant information is communicated in parallel to a hierarchy of system data processing terminals, e.g., those located at area, branch and individual work station locations. Storage media at the several system data processing levels extracts and stores data base information of differing purport and completeness for the disseminated data to support the system work station users.

Abstract Text (2):
In accordance with one aspect of the present invention, information characterizing a dynamically changing sub-population of the overall ensemble of market securities is maintained at and becomes immediately available to each work station responsive to the pattern of usage at that specific station. Various derivative tasks, such as security price limit alerts, are user programmable and are activated by the contents of the work station data base.

Brief Summary Text (5):
It is another object of the present invention that stored program controlled subscriber work stations in a financial market information communication and display system permit local and immediate access to a dynamically changing sub-population of securities of particular interest; and that full securities data is stored on a hierarchal basis at varying system facilities.

Brief Summary Text (6):
The above and other objects of the present invention are realized in a specific, illustrative system for distributing, processing and displaying financial market ticker, quotation, news and ancillary information via a plurality of stored program controlled work stations. Stock trade executions, quotations and other ticker plant information is communicated in parallel to a hierarchy of system data processing terminals, e.g., those located at area, branch and individual work station locations. Storage media at the several system data processing levels extracts and stores data base information of differing purport and completeness for the disseminated data to support the system work station users.

Brief Summary Text (7):
In accordance with one aspect of the present invention, information characterizing a dynamically changing sub-population of the market securities is maintained at and becomes immediately available to each work station responsive to the pattern of usage at that specific station. Various derivative tasks, such as security price limit alerts and customized, selective ticker displays, are user programmable and are actuated by the work station data base.

Drawing Description Text (4):
FIG. 3 is a flow chart illustrating user work station data processing to generate quotation information and to dynamically update the work station data base market

security sub-population; and

Detailed Description Text (2):
Each work station 110.sub.i,j,k has access to information stored in more senior computers in the computer hierarchy of the instant invention. Thus, for example, the broker at the illustrated work station 110.sub.i,j,k (and all others similarly situated) has access to his branch computer 90.sub.j,k and, in particular, to the variable content RAM memories 95 and 96 there located which supply information beyond that capable of storage in the RAM 111 of work station 110.sub.i,j,k. Yet further continuing up the computer hierarchy, the work station 110.sub.i,j,k has access to the contents of a RAM 60 in an area-serving computer 50.sub.k with which its branch is associated. Ultimately, all system work stations 110 can access the master customer data base memory 12 in a home office main frame computer.

Detailed Description Text (5):
Advantageously for market information continuity assurance, the receiving location apparatus 70 includes antennas 80 and 81 for respectively receiving each of the satellite and television radiated signals. Examining the receiving equipment shown in FIG. 1A for area computer 50.sub.k, illustrative of all such apparatus, the satellite and VHF or UHF television-multiplexed signals are respectively received at antennas 80 or 81 and detected by RF receivers 78.sub.1 and 78.sub.2. Antenna surrogates, such as cable television delivery systems, may be employed. A demultiplexer 77 selects the base band data stream output of one or the other of radio receiver/detectors 78.sub.1 or 78.sub.2 under control of central processor 72 in accordance with any appropriate algorithm stored in a ROM memory 74. Thus, for example, the CPU can receive and temporarily store in a RAM 76 the data stream outputs of both receivers 78.sub.1 and 78.sub.2 and select that one exhibiting the lower error rate. Other selection algorithms will be readily apparent to those skilled in the art.

Detailed Description Text (8):
As a matter of overall system philosophy, when a work station 110.sub.i,j,k seeks current price information for a security not then within its memory 111, it seeks such information from its associated branch computer 90.sub.j,k. If the information is not available at the branch level, the branch computer 90.sub.j,k inquires of the area computer 50.sub.k via connecting modems 91 and 52 and communications link 83. Thus a reasonable amount of memory and computing power is employed at the several system hierarchal levels commensurate with the reasonable needs of those levels. All information is obtainable at a work station 110 either from its internal storage, from its branch, or from its area. Additional information may be obtained, as needed, from the home office main frame central processor 14 and data base 12 (source level 10) via communications apparatus 22 (e.g., the switched telephone network) and modem 25; or from external data base(s) 26 via a modem 27.

Detailed Description Text (12):
One illustrative multi-window presentation for display 107 at a system work station 110 is shown in FIG. 2. The composite presentation has a first field 142 which simply comprises the complete New York Stock Exchange ticker (a series of stock transaction messages for stock executions on that exchange). The field includes a sequence of messages each formed of a stock symbol 143 followed by the volume (in hundreds of shares) 144 and the trade price 145. The price 145 may have its first digit deleted, and volume may be omitted on reasonably busy days to obviate undue ticker delays. Examining, for example, the first trade constituent in the ticker data field 142 in display 107, one viewing the ticker would know that 5,000 shares of the security having an exchange symbol ABC traded at a price of 903/4.

Detailed Description Text (15):
A further, MONITOR field 153 contains price information for a predetermined population of securities of interest to that particular broker. Each entry includes an identification 154 (the stock symbol), a designation 155 of the market where the

last trade occurred, an arrow 157 signalling whether the last trade was an uptick or downtick, and the currently obtaining bid and ask prices 158. Finally, a QUICK-QUOTE field 157 provides a quotation for a particular stock (ABC) having a symbol entered by the user via keyboard 112. Reading across the illustrative entry of FIG. 2, the stock symbol is followed by an identifier for the exchange executing the last trade (New York) in the security, an arrow showing the tick direction of the last trade (up), the trade price (903/4), the current bid (905/8) and asked (907/8) prices and the exchanges where those bid and ask prices came from (bid-Boston and New York, asked-American and Toronto), the bid and ask volume sizes (60 and 5 respectively), the number of shares (230,800) of that security traded so far during that business day, and the time of the last trade (12:02).

Detailed Description Text (19):
As new trades in the monitored 300 security population are reported via the ticker plant 35, communications link receiver 98, demultiplexer 105 and work station central processor 103 automatically change the stored price information in RAM 111. The stored security price information also automatically changes the price presentation for the respective securities wherever a security appears in any of the multiple windows (fields) of the display 107. That is, new price information for any particular stock will change in each window in which that security appears. For example, a price change in the price of the equity whose symbol is assumed to be ABC for the illustrative display of FIG. 2 causes changes in at least the MONITOR field 153, the NYSE Ticker 142, and in the QUICK-QUOTE field 157 all of which derive their refreshed information from the work station data base in RAM 111. If the last trade exceeded a limit, an appropriate message would be generated as well in field 151.

Detailed Description Text (20):
Attention will now be directed to the flow chart of FIG. 3 which presents the operative program for maintaining the data base in the work station 110 RAM 111 to reflect the limited (300) entries most recently queried at that specific work station. The program for dynamically controlling storage at the user's work station 110 is typically stored in the read only memory, or ROM 109 there included although RAM 111 storage is also possible. To reiterate, it is the function of the dynamic storage algorithm to maintain in the variable, RAM memory 111 at the user's station 110 information associated with the 300 securities for which quotations were most recently requested at that station (and which are thus most likely to generate future quotation requests). To this end, RAM 111 includes a "least recently used" or LRU list which stores the stock symbols of the most recently requested 300 (or fewer) securities. It will be assumed that the most recently requested security resides in the top, or first position, in that list; and that the least recently requested symbol is stored at the bottom of the list in a jeopardy position to be purged if a new security, not otherwise in the LRU list, is entered at the work station keyboard 112 (assuming a full complement of 300 items). A processing variable LRUSZ is maintained to indicate the size or number of items in the LRU list. Programming for the instant invention may of course be in any convenient language which is stored and implemented on any of the diverse forms of digital processing apparatus.

Detailed Description Text (21):
To illustrate specific operation of the dynamic storage reallocation algorithm, assume that a broker or other user at the work station 110.sub.i,j,k illustrated in FIG. 1B wishes a quotation on any desired security. He enters the corresponding symbol for the security as by his signal entry keyboard 112 (functional step 201 in FIG. 3). Test 205 then examines the LRU table to determine whether the newly entered stock symbol is already in the LRU list. If it is (YES output of test 205), test 206 examines the command message entered through keyboard 112 to determine whether the user wishes a full quote (e.g., including historical and derived (e.g., price-earnings ratio) information not locally available at the work station 110 or the more common so-called quick quote price and volume information which is locally

available. If a full quote is desired, the work station 110 obtains the historical information from the historical information memory 95 in the branch computer 90.sub.j,k via the communicating demultiplexer 105. If desired, historical information of varying levels of detail may be distributed between the branch and area RAMs 95 and 60. In either event, either the quick quote or full quote after data retrieval from RAM 95 is displayed for the user (step 215) as via the user's cathode ray tube display 107.

Detailed Description Text (22):
Tracing the alternate output path from the test 205, assume that the stock symbol requested was not one recently examined at the specific work station 110.sub.i,j,k and therefore was not in the LRU list locally available from the work station 110 RAM 111 (N.0. output path of test 205). When this condition obtains, the desired quotation is retrieved from the branch RAM 96 (or higher order computer if necessary)--step 220. Depending upon whether a full quote or quick quote was specified by the input command entered by the user at keyboard 112, test 222 fetches the full information from the branch RAM 95 if appropriate (step 224) or skips this operation if only a quick quote was desired. The following operation 227 stores the securities information just obtained in the user's work station variable memory 111, and step 229 sets a flag bit in some predetermined location (e.g., FLAG) to signal that FIG. 3 processing is dealing with a security not previously stored at the user's work station 110 memory 111. As before, the quotation information is displayed in its full or quick (limited) form in the display step 215.

Detailed Description Text (24):
The remainder of the functional operation depicted in FIG. 3 then serves to maintain the least recently used (LRU) list in correct form as well as to maintain the list size variable (LRUSZ) at the correct value. To this end, test 219 examines the contents of the flag bit (FLAG) to determine whether or not the symbol most recently processed was new to the data table (it being new following the N.0. output of test 205 but not for the YES output of that test). If the flag bit was not set (N.0. output of test 219) signalling that the stock symbol (and its concomitant information) was already in the LRU list and in the RAM 111 data table, step 230 searches through the LRU list after position 1 and deletes the second appearance of the symbol in the list. The symbol is deleted since it is known to be in the first or most senior position in the LRU list as a result of step 217 and thus its redundant presence is discarded. That completes operation of the FIG. 3 dynamic storage reallocation for the assumed branch of data processing which thus goes to the end point of the subroutine and passes to system control for other system business.

Detailed Description Text (26):
Accordingly, the FIG. 3 mode of data processing automatically maintains within the work station 110 RAM 111 a list (LRU) of the 300 most recently requested stock symbols at that station. The newer of the stock symbols requested are in the top portion of the list while the older symbols are in the bottom part of the list, with symbols being deleted if they are not requested a second time before 300 other quotations are entered at the user keyboard 112.

Detailed Description Text (27):
It will be apparent that the stock symbols in the LRU list and the corresponding stock price values and other information, will vary from time to time for any user of the equipment 110.sub.i,j,k and will differ at any given time for different system work stations presumably having operators who enter different patterns of quotations. The local variable memory 111 of each work station 110 will thereby store the information most likely to be next needed by each station user and which will be quickly available to that person, not requiring interrogation (other than for "historical" information) from any other system computer thus obviating communication and possible queueing delays.

Detailed Description Text (28):
Finally, attention will be directed to the flow chart of FIG. 4 which presents the operative program for dynamically updating data in the user's RAM 111 data base characterizing the stocks having a present application for that user. That is, FIG. 4 depicts the manner in which current price and other market data is loaded into the user's RAM 111 to provide current information for each component of the display (FIG. 2) of the user's cathode ray tube 107. It will be assumed for simplicity of discussion that each separate display application (FIG. 2 field or window) has an associated list in RAM memory 111 of those symbols currently of interest, i.e., there exists a first list (LRU table) for the 300 most recently requested quotations, a second list for those securities for whom limits are being maintained, further lists for the ticker presentations, and so forth. Each list would contain or have a pointer to all data for each security in that list. Alternatively, a single integrated list and data table may be employed for all stocks for which there is any current application, together with one or more identifiers which record those application(s) for which the stock data is required.

Detailed Description Text (29):
Examining the flow chart of FIG. 4, the first step 301 reads into the computer CPU the next incoming stock symbol, price, volume and related information (ticker message) originated by ticker plant 35, and furnished to the work station 110.sub.i,j,k via its corresponding branch apparatus 70,80,81 via cable 103 and demultiplexer 105. Test 303 examines each of the application stock lists (i.e., the LRU list, the list associated with the limit processing, and so forth). If the security being characterized by the ticker plant message is not in any such list (N.O. output of test 303), control passes to test 320 to determine whether or not the stock data is appropriate for one of the tickers (e.g., 142 or 147 of FIG. 2) in the user display. Assuming that the trade information being reported by ticker plant 35 is germane to one or more of the applications for that specific work station 110, the data base in RAM 111 associated with that security is updated (step 308) to reflect the last trade and quotations for that stock and step 310 updates all applications (windows and the related window-driving storage) associated with that stock as necessary. Thus, as only one example and assuming that the stock having the trade information then being reported by the ticker plant was in the LRU list and data base, the information being reported replaces the older data for that security stored in the data base of the user's RAM 111.

Detailed Description Text (30):
Assuming the stock to be one maintained in the limit table (supporting display field 151 of FIG. 2), test 312 determines whether the trade being reported exceeds any limit bound. If it does not (N.O. output of test 312), system control passes to test 320 for ticker processing. If a limit is exceeded (YES output of test 312), a limit-exceeding message appears in the field 151 of FIG. 2 advising the user of the appropriate circumstances. It will be readily apparent that a price may be tested against upper and/or lower bounds as desired for the investment strategy of the user, or of the customers of the user. In addition, step 317 recalls from the master customer data base 12 via multiplexer 105 the branch modem 91 and all remaining communication apparatus intermediate the data base 12 and work station 110 the name, account number, telephone number and all other desired information for all customers who hold the security for which the user's station has indicated an out-of-limit message. As appropriate the user may contact each such owner of the subject security to determine if any action is desired or to take such automatic action as may be appropriate.

Detailed Description Text (31):
Finally, test 320 examines the subject ticker plant 35 message to determine whether or not it is appropriate under the criteria established by the user at his work station 110 for either of the ticker streams 142 or 147 being displayed. If the

criteria is satisfied (YES output of test 320) the message is added to the appropriate ticker display memory or memories for entry into the appropriate ticker. The ticker criteria as above noted is subject to definition by the user. If the user has limited a ticker to a finite group of stocks, the ticker criteria is satisfied if and only if the stock symbol in the incoming message matches a stored desired symbol. Other criteria will be readily apparent, e.g., to display only trades from a particular exchange (part of the data transmitted by ticker plant 35). If the ticker plant message is not appropriate for display on any ticker (N.phi. output from test 320), control passes to the beginning of FIG. 4 processing to await the next trade quotation being supplied by the master ticker plant 35 (or to shift to other system functions).

Other Reference Publication (2):
Everest, G. C. Database Management. (McGraw-Hill Book Company: New York) 1986, 746.

CLAIMS:

7. A combination as in claim 1, further comprising a customer data base, and communications means selectively coupling each of said work stations with said customer data base.